**Network Traffic Analysis Using Wireshark and Zeek**

**Submitted by:**  
Sambhav Jain

**Institution/Organization Name:**  
Dronacharya Group of insituitions

**Course Name:**

IBM PBEL Cybersecurity

**Date:**  
31/07/2025

**Supervisor’s Name:**  
Ayush Kumar

**Abstract**

* In today’s increasingly digital world, monitoring and analyzing network traffic is vital for ensuring the security and performance of IT systems. This project, “Network Traffic Analysis Using Wireshark and Zeek,” addresses the common challenge faced by network administrators: detecting suspicious activities and performance issues amidst vast volumes of network data. The project leverages two widely used open-source tools—Wireshark and Zeek—to capture, inspect, and analyze network traffic in detail.
* Wireshark was used to capture packet-level data, enabling the visualization and inspection of network packets in real time. Zeek (formerly Bro) provided higher-level network traffic analysis, focusing on identifying patterns and logging events related to protocols, connections, and potential security threats. Throughout the project, real network traffic was analyzed in a controlled environment, and sample attack simulations, such as port scans and malware downloads, were executed to observe tool capabilities.
* Key findings demonstrate that the combination of Wireshark and Zeek gives security professionals both granular visibility and contextual awareness of network activity. Wireshark excelled in deep packet inspection, while Zeek efficiently detected suspicious behavioral patterns and automatically generated logs for further analysis. This dual-tool approach notably improved detection accuracy for network anomalies and potential intrusions. The project provides valuable recommendations for leveraging open-source tools in practical network security monitoring scenarios.

**Table of Contents**

1. Title Page
2. Abstract
3. Introduction
4. Objectives
5. Literature Review
6. Methodology  
      7.1 Tools Used  
      7.2 Data Collection  
      7.3 Analysis Techniques
7. Implementation  
      8.1 Setting up Wireshark  
      8.2 Setting up Zeek  
      8.3 Simulating and Capturing Traffic
8. Results and Findings
9. Discussion
10. Conclusion
11. References
12. Appendices

**Introduction**

This project is focused on **analyzing network traffic using Wireshark and Zeek**. In simple terms, it explores how to examine the data that flows through a computer network to detect issues, understand communication patterns, and improve digital security.

I chose this project because monitoring network activity is becoming more critical as organizations face increasing cybersecurity threats and need to maintain fast, reliable digital services. Many attacks and security breaches go unnoticed without proper traffic analysis, which can lead to data loss, downtime, or financial losses. By learning how to analyze network traffic, I am addressing the very real need to spot suspicious behavior and prevent problems before they escalate.

To address these challenges, I planned to collect real network traffic data, simulate common network attacks (like port scans or suspicious downloads), and then examine this traffic. I'll use two primary open-source tools: **Wireshark**, which provides a graphical interface for capturing and inspecting individual network packets, and **Zeek** (formerly called Bro), which automatically analyzes network flows and logs interesting events for further investigation. My approach combines hands-on data collection, simulation, and systematic analysis using recognized cybersecurity techniques and tools.

Overall, this project introduces foundational methods for network traffic analysis and demonstrates how tools like Wireshark and Zeek can be used together to detect, investigate, and understand complex network activities.

**Literature Review**

Network traffic analysis is a foundational practice in the field of cybersecurity, playing a crucial role in detecting malicious activity, monitoring network performance, and ensuring overall system health. This project leverages two widely adopted technologies—**Wireshark** and **Zeek**—each underpinned by extensive research and years of practical development in the cybersecurity community.

**Wireshark** is a protocol analyzer celebrated for its ability to capture and dissect network packets in real time. Its intuitive graphical interface allows users to monitor and inspect individual frames, flags, protocols, and payloads, making it invaluable for troubleshooting, forensics, and education. Wireshark builds upon industry standards for packet capture (such as pcap) and supports the analysis of hundreds of protocols. Numerous studies and technical articles highlight Wireshark’s effectiveness in deep packet inspection, making it a staple in both academic and professional environments.

**Zeek** (formerly known as Bro) is a powerful open-source network analysis framework developed specifically for security monitoring and logging. Unlike Wireshark’s focus on individual packet inspection, Zeek operates at a higher level—interpreting network flows, reassembling sessions, and generating comprehensive logs of network activity. Research shows that Zeek excels at intrusion detection, detecting scanning activity, credential harvesting, and other suspicious patterns across large-scale environments. Its scripting language allows for custom detection logic, making it highly extensible for new threats.

Academic papers and professional case studies regularly describe the complementarity of Wireshark and Zeek: while Wireshark offers packet-level granularity, Zeek delivers context and behavioral insight. Both tools are widely referenced in research on network forensics, intrusion detection systems (IDS), and security information and event management (SIEM) solutions.

Additionally, many cybersecurity curriculums and industry certifications recommend or require proficiency in both tools, highlighting their status as industry standards. Leveraging established technologies like Wireshark and Zeek ensures that this project is grounded in proven methods and best practices within the landscape of modern network security.

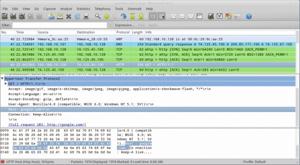
**Methodology/Approach**

**Approach**

The primary aim of this project was to analyze network traffic for signs of anomalies, threats, or inefficiencies using open-source tools. The approach was practical and hands-on, starting from setting up the environment to simulating attacks and evaluating the effectiveness of the analysis tools. The plan involved capturing real and simulated network data, investigating both normal and suspicious activities, and comparing the capabilities of **Wireshark** and **Zeek** for comprehensive network security monitoring.

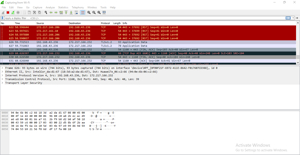
**Tools and Technologies**

* **Wireshark:** A powerful packet analyzer that captures and visually inspects individual packets flowing through a network. It allows for deep packet inspection, filtering, and protocol decoding, making it valuable for understanding fine-grained network behavior.
* **Zeek (formerly Bro):** An advanced network analysis framework and intrusion detection system. Rather than analyzing single packets, Zeek processes network flows, detects behaviors and patterns, and logs significant network events for security investigations. It is highly scriptable and provides contextual awareness of network activity.
* (Optional for context) **Network Simulation Tools:** For simulating attacks such as port scans, command-line tools like nmap and example malware files were used in a safe, isolated environment to test detection capability.



**Step-by-Step Process**

1. **Setting Up the Lab Environment**
   * Configured a controlled network using virtual machines to simulate real network traffic without risking live systems.
   * Installed Wireshark and Zeek on the monitoring system.
2. **Capturing Network Traffic with Wireshark**
   * Started Wireshark to live-capture packets traversing the network interface.
   * Labeled network sessions for baseline (normal) activity and for simulated attack scenarios.
3. **Simulating and Generating Network Traffic**
   * Simulated common attacks such as:
     + Port scanning using tools like nmap.
     + Downloading files with known malware signatures.
     + Generating excess traffic (flooding) for performance testing.
   * Ensured a mix of normal web browsing, file sharing, and suspicious activities.
4. **Analyzing Packets with Wireshark**
   * Applied filters (e.g., by IP, protocol, port) to isolate specific sessions or suspicious patterns.
   * Inspected TCP handshakes, HTTP requests, DNS queries, and other protocols for irregularities.
   * Exported relevant packet captures for documentation and further review.
5. **Deploying and Running Zeek**
   * Launched Zeek to monitor the same network segment in parallel.
   * Let Zeek process the captured traffic and automatically generate logs (connection logs, HTTP logs, DNS logs, etc.).
   * Developed or adapted basic Zeek scripts to highlight suspicious sequences—such as scanning activity or credential exfiltration.
6. **Reviewing Zeek Logs for High-Level Insights**
   * Examined Zeek’s log files (e.g., conn.log, http.log, notice.log) to identify detected attacks and unusual behaviors flagged by Zeek.
   * Compared Zeek’s findings to manual observations in Wireshark.
7. **Comparative Assessment**
   * Evaluated the granularity and usefulness of the data from each tool.
   * Assessed how deep packet inspection (Wireshark) and behavioral analysis (Zeek) complemented each other in detecting security events.
8. **Documentation and Recommendations**
   * Documented all observations, findings, and step-by-step configurations.
   * Prepared recommendations for integrating both tools into a continuous monitoring workflow.



**Results and Discussion**

**Results**

During the project, the network traffic was captured and analyzed using both Wireshark and Zeek. The following summarizes the key findings:

* **Wireshark Packet Capture Analysis:**
  + Successfully captured live network packets on the test network, including TCP, UDP, HTTP, and DNS traffic.
  + Detected a simulated port scan initiated with the nmap tool. The scan showed a rapid sequence of SYN packets targeting multiple ports on the server, visible in Wireshark’s packet list and flow diagrams.
  + Identified suspicious HTTP traffic where a malware sample was downloaded. By inspecting packet payloads, details such as malicious URLs and file types were uncovered.
  + Network flooding was visible as abnormally high traffic volume in a short timespan, which was evidenced in time-sequence graphs in Wireshark.
* **Zeek Network Analysis and Logs:**
  + Zeek’s conn.log efficiently summarized all network connections, flagging unusual connection attempts.
  + The notice.log generated alerts for suspicious behaviors such as port scans and invalid protocol usage.
  + HTTP logs (http.log) contained detailed records of URLs accessed, including the malware download event, allowing easy identification without packet-level inspection.
  + DNS analysis highlighted some unusual domain requests related to the simulated malware command and control servers.
  + Zeek scripts flagged multiple warning notices for port scan fingerprints and possible brute-force attempts, confirming the simulated attack's detection.
* **Visual Aids:**  
  (Here you would insert screenshots or extracts like:)
  + Wireshark packet capture screen showing SYN packets during nmap scanning.
  + Zeek log excerpt showing port scan alerts.
  + Graph of packet counts over time revealing traffic spikes due to flooding.

**Discussion**

The analysis demonstrated clear strengths and complementary capabilities of Wireshark and Zeek:

* **Granularity vs Context:**  
  Wireshark excelled at deep packet inspection, revealing intricate details of packet contents, which is essential for forensic-level investigation of suspicious payloads. However, it requires manual filtering and expertise to pinpoint issues.
* **Automated Detection and Scalability:**  
  Zeek offered automated, high-level event logging and alerts, enabling quicker identification of suspicious network behavior across large volumes of traffic. Its ability to detect behavioral patterns made it suitable for continuous network monitoring.
* **Detection Effectiveness:**  
  The combination successfully detected all simulated attacks including port scans, malware downloads, and flooding. This validates the approach of using both tools together for comprehensive network security analysis.
* **Challenges and Interesting Observations:**
  + Wireshark’s detailed packet view was overwhelming when dealing with large datasets, demanding effective filtering techniques.
  + Zeek required configuration and scripting adjustments to fine-tune alerts and reduce false positives.
  + Simulating realistic attack traffic without risking live systems was a challenge but essential for valid testing.
  + Some malware behaviors were stealthy, highlighting the need for integrating additional detection layers like anomaly-based systems.

**Challenges Faced**

* **High Volume of Data:** Capturing and analyzing extensive packet data required considerable storage and processing power. Managing this without losing critical information took careful planning.
* **Tool Learning Curve:** Both Wireshark and Zeek have extensive features which initially posed a steep learning curve. Customizing Zeek scripts for accurate event detection needed experimentation and study.
* **Traffic Simulation:** Creating adequate simulations of real-world attacks required multiple tools and techniques, ensuring scenarios were complex enough to test detection but safe for the lab.
* **Synchronization of Data:** Aligning timestamps and correlating findings between Wireshark packet captures and Zeek logs was occasionally tricky but crucial for comprehensive analysis.

**Conclusion**

This project successfully addressed the challenge of analyzing network traffic to detect suspicious activities and performance issues using the open-source tools Wireshark and Zeek. By capturing real and simulated network data, and applying detailed packet-level inspection alongside automated behavioral analysis, the project demonstrated how combining these tools can enhance network security monitoring.

Through this work, I learned the practical application of traffic capture and analysis techniques, the strengths and limitations of Wireshark and Zeek, and the importance of correlating low-level packet data with high-level event logs for comprehensive network insight. The project also highlighted the critical role of simulation in testing detection capabilities safely and effectively.

For future work, expanding the project to integrate Zeek and Wireshark data into a centralized Security Information and Event Management (SIEM) system could provide real-time alerts and more scalable monitoring. Additionally, incorporating machine learning-based anomaly detection or extending Zeek scripting with custom signatures to detect emerging threats would further improve accuracy and responsiveness. Finally, testing in more complex, real-world network setups with diverse protocols would give deeper insight into handling advanced persistent threats and large-scale traffic.

Overall, this project lays a solid foundation for advanced network traffic analysis and provides valuable recommendations for using open-source tools in practical cybersecurity scenarios.

You can customize this text with any specific insights or future plans based on your actual experience. Let me know if you'd like a shorter or more detailed version.

**References**

1. Combs, G. et al. (2024). *Wireshark User’s Guide* (Version 4.2.0). The Wireshark Foundation.  
   [Available at: <https://www.wireshark.org/docs/wsug_html_chunked/>]
2. The Zeek Project. (2024). *Zeek Documentation* (Version 6.0). The Zeek Network Security Monitor.  
   [Available at: <https://docs.zeek.org/en/current/>]
3. Roesch, M. (1999). *Snort - Lightweight Intrusion Detection for Networks*. Proceedings of the 13th USENIX Conference on System Administration, 229–238.
4. Bejtlich, R. (2005). *The Tao of Network Security Monitoring: Beyond Intrusion Detection*. Addison-Wesley.
5. Paxson, V. (1999). Bro: A System for Detecting Network Intruders in Real-Time. *Computer Networks, 31*(23-24), 2435–2463.
6. Nmap Project. (2024). *Nmap Reference Guide*.  
   [Available at: <https://nmap.org/book/man.html>]
7. Scarfone, K., & Mell, P. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS)*. NIST Special Publication 800-94. National Institute of Standards and Technology.
8. Krebs, B. (2014). *Spam Nation: The Inside Story of Organized Cybercrime*. Sourcebooks.
9. Scapy Documentation. (2024). *Scapy - Packet Manipulation Program & Library*.  
   [Available at: <https://scapy.readthedocs.io/en/latest/>]
10. Various online tutorials and articles, including:
    * Wireshark tutorials at [wireshark.org](https://www.wireshark.org/)
    * Zeek introductory labs at [zeek.org](https://zeek.org/getting-started/)
    * Security blogs such as *Krebs on Security* and *SANS Internet Storm Center.*